24.04.95

17

PATENT OFFICE

JAPANESE GOVERNMENT

This is to certify that the annexed is a true copy of the following application as filed with this office.

Date of Application: July 30, 1993

Application Number: Patent Application No. Hei 05-189636

Applicant(s): Canon Kabushiki Kaisha

August 26, 1994

Commisioner,

Patent Office Akira Takashima

Issuance No.Hei 06-3043527

24.04.95

(Name of Document) Application for Patent

(File Number) 2589044

(Date of Submission) July 30, 1993

(Addressee) To Commissioner, Patnet Office,
Wataru Asoh

(International Patent Classification) H04N 5/232

(Title of Invention) CONTROL SYSTEM FOR MULTIMEDIA DEVICES

(Number of Claims) 8

(Inventor)
(Address) c/o Canon Kabushiki Kaisha, 3-30-2, Shimomaruko, Ohta-ku, Tokyo
(Name) Fumiaki Takahashi

(Inventor)
(Address) c/o Canon Kabushiki Kaisha, 3-30-2, Shimomaruko, Ohta-ku, Tokyo
(Name) Akira Mamiya

(Inventor)
(Address) c/o Canon Kabushiki Kaisha, 3-30-2, Shimomaruko, Ohta-ku, Tokyo
(Name) Akira Suga

(Inventor)
(Address) c/o Canon Kabushiki Kaisha, 3-30-2, Shimomaruko, Ohta-ku, Tokyo
(Name) Hideaki Kawamura

(Inventor)
(Address) c/o Canon Kabushiki Kaisha, 3-30-2, Shimomaruko, Ohta-ku, Tokyo
(Name) Takashi Aizawa

(Inventor)
(Address) c/o Canon Kabushiki Kaisha, 3-30-2, Shimomaruko, Ohta-ku, Tokyo
(Name) Kenji Hatori

(Applicant for Patent)
(Identification Number) 000001007
(Postal Code) 146
(Address) 3-30-2, Shimomaruko, Ohta-ku, Tokyo
(Name) Canon Kabushiki Kaisha
(Representative) Hajime Mitarai
(Telephone Number) 03-3758-2111

(Agent)
(Identification Number) 100069877

24.04.95

(Postal Code) 146
(Address) c/o Canon Kabushiki Kaisha, 3-30-2, Shimomaruko, Ohta-ku, Tokyo
(Patent Attorney)
(Name) Giichi Marushima
(Telephone Number) 03-3758-2111

(Indication of Fee)
(Method of Payment) Deposit
(Deposit Account Number) 011224
(Amount of Payment) 21000

(List of Submitted Articles)
(Name of Article) Specification 1
(Name of Article) Drawings 1
(Name of Article) Written Abstract 1
(General Power of Attorney Number) 9003707

(Necessity of Proof) Necessary

(Name of Document)

Specification

(Title of the Invention)

CONTROL SYSTEM FOR MULTIMEDIA DEVICES

(Scope of Claim for Patent)

(Claim 1)

A system comprising a plurality of multimedia devices and a control device therefor connected on a network and in which said multimedia devices and said control device are possible to transmit or receive messages and data based on the object-oriented concepts from one to another through said network, wherein said control device is associated with display means and pointing means, whereby, as icons symbolizing said respective individual multimedia devices are displayed by said display means, the user links up one of these icons with another by using said pointing means, when he or she specifies a relationship for inputting or outputting data between the corresponding two of said multimedia devices.

(Claim 2)

A system according to claim 1, further including

a user interface and its display means in order to display the icons symbolizing the multimedia devices on connection in said network and the current links.

(Claim 3)

A control apparatus for a system, said system having a plurality of multimedia devices and a control device therefor connected in a network and arranged so that said multimedia devices and said control device are possible to transmit or receive messages and data based on the object-oriented concepts from one to another through said network, wherein said control device for controlling said multimedia devices is provided with means responsive to commands from the user for selecting a data source and for addressing a destination and checking means for the compatibility of data formats, whereby when the user selects two of said multimedia devices as the source and destination, said checking means determines whether or not the formats of data in said source and said destination are compatible with each other.

(Claim 4)

A system control apparatus according to claim 3, further including means responsive to an output of said data format compatibility checking means for informing the user of the fact that, because no compatibility is obtained between the formats, the data are impossible to transmit or receive

(Claim 5)

A multimedia device for constituting part of a system, said system comprising a plurality of multimedia devices and a control device therefor on connection in a network and permitting said multimedia devices and said control device to transmit or receive messages and data based on the object-oriented concepts from one to another through said network, wherein response means for an inquiry of file formats available to input or output, as transmitted from said control device, is provided in combination with means arranged on response, as there are plurality of data formats available to input or output, to inform said control device of the order of priority of these data formats.

(Claim 6)

A control apparatus for a system, said system having a plurality of multimedia devices and a control device therefor on connection in a network and permitting said multimedia devices and said control device to transmit or receive messages and data based on the object-oriented concepts from one to another through said network, wherein, to control said multimedia devices, said control device has means for introducing data, with which a class is written based on the object-oriented concepts, from the outside, and means receptive of said class for generating an object.

(Claim 7)

A data recording apparatus for a system, said system permitting a plurality of devices to transmit or receive data from one to another through a network, including means operating in such a manner that, as time-serial data are transmitted in real time fashion, when not in data record mode, information of the header message of said time-serial data is to temporarily store, and means operating in such a manner that, when switched to the data record mode, the header message is read out of said storing means and, as preceded by said header message, said time serial data are recorded on a recording medium.

(Claim 8)

A control apparatus for a system, said system permitting a plurality of devices to transmit or receive data from one to another through a network, including detecting means for determine that a program data has been transmitted through said network and actuating means responsive to arrival of said program data sensed by said detecting means for automatically starting execution of this program based on said program data.

(Detailed Description of the Invention)

(0001)

(Field of Utility on Industry)

This invention is suited to be used in controlling a system for multimedia devices of handling many items of information such as letters, voices, sound, still pictures and motion picture.

(0002)

(Prior Art)

In the field of art of audio, video, TV or like audio-visual (AV) instruments where the techniques have so far been developed in respect to analog signals at the core, the recent trend is rapidly shifting to digitization. As information of texts and still pictures in digital form is widespread, it has come to take data of texts, voices, sound, still pictures and motion picture as the so-called multimedia and handle with them unitarily in a computer.

(0003)

(Subjects the invention is to solve)

At present, however, in a case that a multimedia device (digital camera, CD-ROM player, scanner, sound board, video board or like audio input/output device or like video input/output device) is used in combination with the computer, it is necessary to install the software solely dedicated to it, called the "application" or "device driver" in the computer.

(0004)

So long as this method is relied on, therefore, for every new multimedia device to develop, new different

applications or device drivers must be prepared with different types of computers or different OS's (operating systems). Accordingly, the cost of developing the software was huge. There was another problem that an efficient and high speed control was impossible to make.

(0005)

Another problem of the method described above is that, from another computers which were usually connected to a LAN, their users could not transparently use that multimedia device. Hence it has been impossible to realize a concept of multimedia such that every computer can access any of the peripheral devices over the LAN.

(0006)

The subject of the invention is to solve the above-described problems. An object of the invention is to obviate the necessity of using especial software such as the above-described application or device driver in these multimedia devices. Another object is to provide an environment which enables the multimedia devices to use from other controllers transparently. Particularly for a case that transmission of data takes place just between the multimedia devices, provision is made for a user friendly interface so that the user can quickly and easily construct a relationship between the multimedia devices as a data source and a destination therefor. In actual practice, when transmitting data, the controller does not intervene between the paired ones of the

multimedia devices.

(0007)

(Means for Solving the Subject)

The invention has been made to solve the above-described subject. A plurality of multimedia devices and a control device for controlling these multimedia devices are connected in a network, thus forming a system. This system makes it possible that the multimedia devices and the control device transmit or receive messages and data based on the object-oriented concepts from one to another through the network. A feature of the invention is that the aforesaid control device is associated with display means and pointing means. By the display means described above, all the multimedia devices are displayed in the graphical form of respective symbols or icons. Using the pointing means described above, the user selects one of these icons as a data source and addresses another as a destination, thus achieving a linkage therebetween. This is reflected to establish a specified relationship in which the corresponding multimedia devices transmit data from one to the other.

(0008)

(Function)

With this, when to revise the current relationships for transmitting or receiving data between the audio-visual devices, there is no need to change the physical configuration of wiring. Instead, all what is

needed to realize the control system described above is only update the internal data of those objects which have participated in establishing a new relationship for transmitting or receiving data, or revising one of the current relationships.

(0009)

For every object, it is no longer necessary to make preparation of a control program therefor and install it in the controller. So, all what is needed to realize formation of control means for the object is only connect an audio-visual device to the controller through the network.

(0010)

For the controller, as the control means described above is sent from every one of the connected objects, because it is the human being who actually gives "control" commands, it displays all the articles of control means and lets him or her manipulate them. On one interface, therefore, it becomes also possible to manage all the audio-visual devices connected to the network.

(0011)

Also, within the controller, means is provided for monitoring the current connections. This monitoring means is made to cooperate with the display means described above to illustrate the status of the network. Therefore, it becomes also possible that even if there are a great number of current connections, the user can

quickly and easily perceive the states of the current connections at a glance.

(0012)

The present invention is next described in connection with embodiments thereof in great detail by reference to the drawings.

(0013)

At first, in the invention, the multimedia devices are grasped as individual objects, and use is made of a system control technique that the controller unitarily manages these objects.

(0014)

For all the objects, to permit them to be controlled individually by the controller, they have a common function of sending their own specific functions and control means to the controller. This leads to obviate the necessity of previously installing the control program in the controller as was conventional. Thus, all what is needed to realize a control system for all the devices is only connect them to the controller.

(0015)

Again, the controller is provided with means for displaying the control means described above as sent from the connected object and for letting them be manipulated by the human being who actually give control commands. By this, it becomes also possible for the controller to centrally control the multimedia devices. Further, even for a new multimedia device to develop

later, it is possible to cope with it without having to make any new preparation, thus realizing achievement of a great increase of flexibility and versatility.

(0016)

Incidentally, the object-oriented concepts used in the invention are themselves described in detail in the documents for reference, for example, Ishizuka: "Object Oriented Programming" ASCII Publishing 1988; Sakai: "Introduction to Object Oriented Technique" Ohm Co. 1990; and B.J. Cocks: "Object Oriented Programming" Toppan 1988. In the following, therefore, the embodiments of the invention are described with omission of the fundamental techniques.

(0017)

This orientation to objects has been attracting the spotlight of attention from the standpoint of the recent trend of improving the efficiency of program development environment. Moreover, this technical idea can be widely applied even to the OS's and multimedia database. In particular, the characteristic concepts of the object orientation are:

(1) Encapsulation;

(2) Inheritance; and

(3) Messaging

With these three points as the basic concepts, the present invention has made attempts to develop and expand the technology so that it can be applied to making control of multimedia devices.

(0018)

Another technique used in the invention is that the audio-visual devices and the controller are connected in a network so that, as these audio-visual devices are grasped individually as objects, the controller manages these objects unitarily. Every audio-visual device, when connected in the network, is given an object ID which is used as an address of destination when messages and data based on the object orientation are to transmit. Based on this object ID, selection of a destination device to which data is to transmit is logically determined. So, to interchange between any two of the audio visual devices to establish, or reverse, a new, or the current, relationship for transmitting or receiving the data, there is no need to change the physical configuration of wiring. All what is needed to do is only update the internal data of the corresponding two of the objects.

(0019)

Another feature is that every object has to be controlled by the controller and, therefore, is given a function of sending the functions and control means it posses in itself to the controller. By this, it is made no longer necessary to prepare a control problem beforehand in the controller as was conventional. It is thus made possible to realize a control system only by connecting the device to the controller. The controller is associated with means for displaying all the control

means as sent from the objects when connected, and for permitting the control means to be manipulated by the human being, since it is he or she who actually gives commands controls. This leads to provide a possibility of only one interface sufficing for permitting manipulation of all the current audio visual devices in the network. Further, the controller is made to include means for monitoring the sorts and status of links between the devices. This monitoring means cooperates with the aforesaid display means to illustrate the sorts and status of the current links between the devices. Therefore, even if a great number of links are constructed in the network, the user can quickly and easily recognize the status of the network.

(0020)

Yet another feature is that using the input or pointing means described above, the user can edit the current connections. This result is transferred by messaging to the monitoring means described before. Having understood the intention of the user, the monitoring means then accesses that object which keeps knowledge on the input and output data, then requests it for information about data to input or output, then check to see if the formats of data between the source and destination devices are compatible with each other and then displays the result on the display means. Therefore, the user can easily know whether the data is possible or impossible to transmit or receive between

the devices.

(0021)

A furthermore feature is that the display means described above displays all the audio-visual devices in the form of respective individual objects symbolic thereof. So the user draws lines pointing from some of the objects to others by the pointing means described above. In such a manner, new relationships between the source and the destination are established. Therefore, there is no possibility of occurrence of the user making any error connections.

(0022)

Fig. 1 shows a logic form of connection of a multimedia controller with multimedia devices employing the object-oriented concepts of the invention. The multimedia controller 1 at the center is surrounded by a number of multimedia devices 2 with their respective individual communication paths to the multimedia controller 1 being established so that direct conversation of various items of information can be made between the multimedia controller and the multimedia devices in one to one basis. It will be appreciated that the control is made by transmitting messages to each other over that communication path. The term "multimedia devices" herein used means, specifically speaking, CD players, digital VTRs, digital cameras, digital TV sets and other AV devices, and digital FAX, digital copiers, printers and other OA devices, that is, all of those devices which

deal with multimedia data.

(0023)

The controller is assumed here to be part of hardware dedicated solely to this purpose. But it is also possible to realize an equivalent controller by installing an especial OS and a particular application on the commonly available processor in the personal computer or word processor.

(0024)

Referring next to Fig. 2, there are shown three configurations (a) to (c) for physically connecting a multimedia controller to a number of multimedia devices to establish the respective duplex communication paths.

(0025)

The daisy chain line of Fig. 2(a) is employed in SCSI bus (ANSI X3.131-1986). The star configuration of Fig. 2(b) is employed in Ethernet (IEEE 802.3) 10BaseT. The multipoint line of Fig. 2(c) is employed in Ethernet 10Base2/5.

(0026)

It is also to be noted that with regard to another possible configurations, there is GPIB (IEEE 4888) as obtained by mixing the (a) to (c). Even in Ethernet, the (b) and (c) may be mixed. With regard to another possible communication systems, there are optical fiber cables and ISDN. So it is to be understood that, besides those of Fig. 2, many other combinations are possible to make and may be selectively employed as desired.

(0027)

How to establish such duplex communication paths and which to select are not essential to the invention, so no particular remarks are given except that, as the communication system differs from one to another, some physical limitations are laid on the transfer speed, the number of connected devices, the length of the cable, the shape of the connector, etc. For the each other's transmission of messages, because the protocols have their hierarchies differentiated from each other, these limitations are out of question. In order to insure that the peripheral devices each are connected to the controller reliably and accurately, however, there is need to provide for the system with at least one physically (mechanically and/or electrically) common interface.

(0028)

To realize high speed transmission of data such as those of motion pictures, it is recommended to employ what is faster than Ethernet, that is, the optical transmission such as FDDI (Fiber Distributed Data Interface) or B-ISDN. But, in here, for the purpose of simplicity of explanation, discussions are conducted on assumption that Ethernet 10Base2(/T) which, because of its cheap price, is widely used is adopted as the common communication connector.

(0029)

The internal pieces of hardware of the usual

multimedia device are shown in an block diagram of Fig. 3.

(0030)

A plurality of multimedia devices are connected via a LAN 4 to the controller. Now this LAN is Ethernet so each of these devices is provided with an interface 20 for implementing its protocol (TCP/IP). This can be realized by using an exclusive LSI or the like. It is in here that the transmitted message itself is taken out. Conversely it is from here that a message is sent out to the controller. An example of these messages, if in the Objective-C, is given, as the general format, by the following expression:

(0031)

(Terminal Object; Method Name; Parameter)

In another languages, the expression takes different styles, but is similar in the following basic components:

(0032)

(1) Addressing a terminal object;

(2) Selecting a method (instruction to execute); and

(3) Putting data in parameters, if any.

How to deal with this message is described in connection with the flow of software of Fig. 33.

(0033)

In the interior of the multimedia device, a

CPU 11 processes all software and controls all hardware through an external bus 10. The programs, the initial values and proper data are stored in a ROM 12. To temporarily store data and internal parameters such as those representing the device status, there is a RAM 13. When executing the programs, this RAM 13 is used as a work area. A data I/O 14 is used in accessing multimedia data stored on an internal or external medium 15. A mechanical system driving portion 16 controls mechanical parts 17 such as an electric motor. An electrical system driving portion 18 controls electrical circuits for switches SW and indicators such as LEDs. As the multimedia data are of digital form and range from pictures to sound to texts, it is possible for the medium 15 to take various types, namely, optical disks such as CD-ROM and MD, magnetic tapes such as DCC and DAT, and semiconductor memory cards.

(0034)

Referring next to Fig. 4, a block diagram shows the internal hardware aspect of the multimedia controller 1. Connections to the multimedia devices are established via the LAN 4. Now this LAN 4 is Ethernet, so there is an interface 31 for implementing its communication protocol (TCP/IP). This can be realized by using a LSI or the like solely dedicated thereto. It is in here that the transmitted message itself is taken out. Conversely it is from here that messages are sent out to the multimedia devices.

(0035)

In the interior of the multimedia controller 1, a CPU 21 processes all software and controls all hardware through an external bus 30. The programs, the initial values and proper data are stored in a ROM 22. To temporarily store data and internal parameters such as those representing the device status, there is a RAM 23. When executing the programs, this RAM 23 is used as a work area. A multimedia filing device 25 performs storing, retrieving, reproducing or editing of multimedia data, regardless of whether filing device is an internal or external medium. Accessing to it is controlled by a data I/O 24. An electrical system driving portion 28 controls electric circuits for switches SW and LEDs or like indicators. A display 27 constitutes a man-machine interface. Its displaying operation is controlled by a controller 26. There is further included a mouse or like pointing device, though not shown.

(0036)

Fig. 6 is a diagram of the hierarchy of systems in the software aspect of the multimedia device. The internal block diagram of Fig. 3 refers to hardware 57. An OS 58 is fundamentally in charge of this hardware. What type of OS to select is not itself particularly limited, but it is desired that the real-time facility and the multitasking capabilities that run more than one program in parallel at a time are available in combination. On this OS, the multimedia device has a

class library 59 which is differentiated from the other devices in order to realize embedding of an object into that multimedia device.

(0037)

Though not shown, the multimedia device has another library concerning the control panel for itself and the control program. When connected to the multimedia controller, this library is transferred thereto, thus permitting a specific control to the multimedia device to be made at the side of the controller. There is also a C function 60 to be used as a timer and in performing arithmetic computations.

(0038)

At the top of the hierarchy, there is an application software 61 which takes its part in controlling the main system of the multimedia device and communicating with the multimedia controller, and as the user interface. The use of this application makes it possible that the main system of the multimedia device as one object is controlled in a variety of ways by transferring messages from or to the controller, and that the internal parameters are read as instance variables and altered.

(0039)

Fig. 5 is a diagram of the hierarchy of systems in the software aspect of the multimedia controller. The internal block diagram of Fig. 4 refers to hardware 50. An OS 51 is fundamentally in charge of this

hardware. In here, too, what type of OS to select is itself not particularly limited. But it is desired that the real-time facility and the multitasking capabilities are available in combination.

(0040)

On this OS there is a window server 52 which is in charge of the whole of a GUI (Graphical User Interface) that displays a plurality of control panels for the connected multimedia devices and the status of all links in the network on the screen, and controls coordination of inputting and outputting of data. A common class library 53 stores what has been made ready beforehand in the controller by itself, that is, a set of basic and common components (in the form of objects) concerning the user interface, namely, buttons, slide volumes and text presentation areas and also concerning the control.

(0041)

For every multimedia device, on the other hand, a specific class library 55 stores a set of components (in the form of objects) concerning the unique panel display and control. This specific library, as described before, increases its content each time one more multimedia device is brought into connection with the system, as an additional set is sent from that device. This procedure will be described later in more detail. There is also a C function 54 for the timer and arithmetic computation. At the top of the hierarchy, there is an

application software 56 which undertakes in controlling coordination of all the connected multimedia devices and communicating with these multimedia devices and that functions as the user interface.

(0042)

The flow of control signals and the transmission of messages between this controller and the multimedia device are described below.

(0043)

Fig. 7 shows a state of the system before the multimedia device is connected to the multimedia controller. In Fig. 7, digital data are transmitted over a communication line or LAN 4. A multimedia controller 1 controls operations of all parts of the system. A multimedia device 2 to be connected to the LAN 4 is shown with its structure of construction in generalized form. 205 is the one of software objects (hereinafter abbreviated to "objects") which always resides in the multimedia controller 1 and coordinates all parts of the system, named "system director".

(0044)

1064 is an object which functions as a multimedia device and which is differentiated from the other objects on the LAN 4, named "multimedia device". This object is further comprised of three objects 1065, 1066 and 1067.

(0045)

The "controller for multimedia device"

object 1065 is in charge of hardware to realize a majority of functions of the multimedia device 2. The "data input to multimedia device" object 1066 is used for entering digital data as transmitted from the other devices over the LAN 4. The "data output from multimedia device" object 1067 is used for transmitting digital data to the other devices over the LAN 4.

(0046)

When the multimedia device 2 is connected to the multimedia controller 1 via the LAN 4, an object that stands in the place of that multimedia device 2 must be formed in the multimedia controller 1. To describe this "deputy multimedia device" object, a specification is written in a file 1061. This file comprises a section 1062 for the specification of a control panel for the multimedia device 2 and another section 1063 for the specification of an input or output of data to or from the multimedia device. In particular, the section 1062 for describing the "control panel for the multimedia device" object realizes the function of describing the control panel so that one interacts with the multimedia device 2 by means of a GUI, that is, the function of the language in which to describe the GUI.

(0047)

Fig. 8 is a diagram to explain the state of the system after the multimedia device 2 is connected to the LAN 4. In Fig. 8, what is now formed in the multimedia controller 1 is an object 1068. In the interior

of the multimedia controller 1, this object acts as substitute for that multimedia device 2, so it is called "deputy" multimedia device. This object 1068 comprises an object 1069 which functions as the control panel for the multimedia device 2, named "control panel for multimedia device", another object 1070 which, when to input data, functions as substitute for the "data input to multimedia device" object 1066, called "deputy" object, and still another object 1071 which functions also as substitute for the "data output from multimedia device" object 1067, called "deputy" object.

(0048)

Fig. 9 is a diagram of the structure of the common class library. For the objects having a similar feature, their common attribute and function have to be defined. To this purpose, a class, say a first class 1079, functions as a template. The first class 1079 to the p-th class 1085, totaling p classes, are summed up in a library 1086. This is conventionally called "class" library. All the objects belong to the respective specified classes. The type and name of data for the internal variables and the type and name of data for the internal functions representing the data processing means (usually called "class methods" the one of the objects which belongs to a particular one of the classes should possess are defined in a portion 1080. To allow one to access to the class methods, all the codes of the class methods are cited in the form of a table 1081 relative

to a pointer. A first function code 1083 through a k-th function code 1084, totaling k function codes for the k class methods, are stored in a "code" portion 1082.

(0049)

Fig. 10 is a diagram of the structure of a typical object. In Fig. 10, an object 234 comprises a portion 244 for accommodating the pointer that is to go to the class method table, communicating means 245 for messages, processing and retrieving means 246, a portion 239 for methods and a portion 235 for internal data. The "method" portion 239 comprises beginning with first data processing means 240, second data processing means 241 and so on and terminating at m-th data processing means 242, totaling m data processing means. The portion 235 for internal data comprises beginning with a first internal data 236, a second internal data 237 and so on and terminating at an n-th internal data 238, totaling n internal data.

(0050)

All the internal data in the portion 235 differ with the different objects and, therefore, are left as they stand in the interior of each of the objects. The data processing means in the "method" portion 239, on the other hand, can be used in common by the other objects, if of the same class. To assure this, therefore, a class method table 243 is provided so that the first data processing means 240 through the m-th data processing means 242 are made manageable by each of the classes.

Thus, such common data processing means are shared by a number of objects which belong to the same class. To look up the class method table 243 from every object, the pointer is brought into this table from an accommodating portion 244 therefor.

(0051)

The message communicating means 245, when in receiving a message from another object, transfers it to the processing and retrieving means 246, where the message is analyzed to identify its address section and the corresponding one of the data processing means to it is retrieved from the "method" portion 239 (virtually the class method table 243). As the selected data processing means operates, the data section of the message, the internal data from the portion 235 and some external data are processed in a predetermined way. In some case, such processing will result in production of a message. If so, this message is transmitted from the communicating means 245 to that other object.

(0052)

Fig. 11 is a diagram of the structure of the system director object 205. A space 1072 accommodates a pointer to a class method table 1073. 1047 is an object forming means for producing the "deputy multimedia device" object 1068 based on the information from the file 1061. Checking means 343 determines whether or not the formats of input or output data between the objects are compatible with each other. Another object forming means 380

produces various application objects of different aims. Further included are communicating means 1074 for messages, processing and retrieving means 342, a "method" part 1075 and an "internal data" part 1076. The internal data are an object ID 1077, control data 344 to be used in establishing links between any two of the multimedia devices when to carry out certain operations, and object record data concerning the connected multimedia devices or the produced objects therefor.

(0053)

When a multimedia device 2 is connected to the LAN 4, using the "deputy multimedia device" object forming means 1047, the system director object 205 reads the file 1061 for describing that object. From the information obtained from this file 1061, it then determines which class the object that should be produced belong to. Based on the definition in the portion 1080 of the corresponding class in the class library 1081, the "deputy multimedia device" object 1068 is made up.

(0054)

Fig. 12 is a diagram of the structure of the "control panel" written section of the "deputy object" written file. In Fig. 12, the section 247 for describing a control panel object comprises first to i-th databases 248 to 249 to be used in describing i articles of the object. One database consists of data 250 for recognizing the object, data 254 for drawing the object and data 260 for object linkage.

(0055)

The data 250 for recognizing the object are of a name 251 of the class which the object belongs to, a unique ID 252 to the i-th article, and an ID 235 of the object to which the i-th article is appendant.

(0056)

The data 254 for drawing the object are used to depict a button or like object constituting part of the control panel window 231 on the screen, comprising first to j-th packs 255 to 259 of data for drawing j articles of the object, One pack consists of data 256 for the location and size at and to which to depict the object, data 257 for the pattern and color and an object image 258.

(0057)

The object link data 261 provide information about the links in the network and are used when one of the items constituting the control panel, for example, the controller object 207, is to connect to another, say relational object, comprising first to k-th data 261 to 264 for establishing k links between the items. One data are of an ID 262 of a relational object, and an message 263 for transmission to the relational object.

(0058)

Fig. 13 is a diagram of the structure of a "deputy data I/O object" section in the "deputy object" file. In Fig. 13, the "deputy data I/O object" section 650 includes first to m-th data 651 to 652 for

producing m articles of a deputy input object. Each data are of an ID 652 of its own article, an ID 653 of a link terminal "data input" object and a compatible file type list 654. The section 650 further includes first to n-th data 659 to 663 for n articles of a deputy "output" object. Each data are of an ID 660 of its own article, an ID 661 of a relational object, and a compatible file type list 662.

(0059)

Next, taking an example of a digital VTR to which the above-described system control method of the invention is applied, the operation of the control system for multimedia devices 2 is described below.

(0060)

Fig. 14 is a diagram showing the state of the control system before an object embedded digital VTR is connected to the multimedia controller. In Fig. 14, as an object is embedded into the digital VTR 203, this "digital VTR" object 206 always resides in the digital VTR 203 and functions as the object embedded digital VTR as viewed from the other devices on the LAN. The digital VTR object 206 is further constructed from three objects. Of these, a "digital VTR controller" object 207 controls the hardware of the digital VTR 203.

(0061)

Another or "data input to digital VTR" object 208 is used to input digital data as transmitted from the other devices over the LAN 4. The other or "data

output from digital VTR" object 209 is used to output digital data for transmission to the other devices over the LAN. When the digital VTR 203 is connected to the multimedia controller 1 via the LAN 4, a deputy digital VTR object is generated in the interior of the multimedia controller 1 based on the information from a "deputy digital VTR object" written file 210.

(0062)

This file 210 comprises a section 211 in which the specification of a control panel for the digital VTR 203 is written, or which is used in describing a "control panel for the digital VTR" object and another section 212 in which the specification of an object that acts as substitute for the data input/output of the VTR 203, or which is used in describing the "deputy data I/O of digital VTR" object.

(0063)

Fig. 15 is a diagram of the structure of the VTR controller object 207, where a portion 1009 accommodates a pointer that goes to a class method table 1018. This table is formed with a wide variety of data processing means including reproducing means 1019 for operating the play mode under the control of the hardware of the VTR and recording means 1020. 1010 is communicating means for messages. 1011 is processing and retrieving means. Though a "method" part is shown at 1012, it is in actual practice that the data processing means are presented by a class method table 1018. The

internal data in a portion 1015 are of many variables and status information necessary to control the digital VTR, for example, the tape running speed 1016 and the current tape footage 1017.

(0064)

Let us first explain an operation which occurs when the digital VTR 203 is connected to the LAN 4. Fig. 16 is a flowchart of the routine for this operation. Fig. 17 is a plan view of a window for the multimedia controller 1 on the screen. In Fig. 17, the multimedia controller window 228 contains a number of icons of which the icon 229 appears when the digital VTR 203 is connected to the LAN 4. Using a mouse or like pointing device, one can choose a location with a cursor 230. The pointing device, though not shown, is provided with buttons. The user presses the button and then releases it. This pressing once is usually called clicking. Pressing it twice in quick succession is called double clicking. Incidentally, as other usable devices, mention may be made of a camera (for inputting still pictures), a tuner, a television set, various relational databases and a CD. To allow the user to select these options, the window 228 displays their icons.

(0065)

Fig. 18 is a diagram to explain the state of the system when the object embedded digital VTR 203 as an example of the multimedia device is connected to the LAN 4. In Fig. 18, an object 220 is now formed in the

interior of the multimedia controller 1. This object serves as substitute for the digital VTR 203, being named "deputy digital VTR". The deputy digital VTR object 220 is constructed from a deputy "control panel for digital VTR" object 221 which functions as a control panel for the digital VTR 203, another or deputy "data input to digital VTR" object 222 which, when inputting data, functions as substitute for the data input object 208, and yet another or deputy "data output from digital VTR" object 223 which functions also as substitute for the data output object 209.

(0066)

Referring now to Fig. 16 and Fig. 17, the routine for such an operation is described below. When the digital VTR 203 is connected to the LAN (636), the system director object 205 detects establishment of a new connection (637). Then it sends a device ID to the digital VTR 203 (638).

(0067)

Using the deputy multimedia device object forming means 1047, the system director object 205 then loads the "deputy digital VTR object" file 210 from the digital VTR 203 (639). Based on the information from the file 210, the system director object 205 then generates a deputy digital VTR object 220 in the interior of the multimedia controller (640) by using the deputy multimedia device object forming means 1047. Such a procedure results in a change of the status of the network

as shown in Fig. 18. Then, the deputy digital VTR object 220 presents the display of an icon 229 for the digital VTR 203 in the multimedia controller window 228 (641). After this, the system stands by for instructions from the user (642).

(0068)

Subsequently, with the help of the window displayed on the screen based on the "control panel for digital VTR" object of the multimedia controller, the user will activate the digital VTR. On this, the control system can operate the digital VTR through the intermediary of the deputy digital VTR object 220 in the multimedia controller 1.

(0069)

Next, the relationship between the content of the "deputy digital VTR object" written file 210 and the object to be generated is described in detail below.

(0070)

Fig. 19 shows an icon for the VTR 203 and Fig. 20 shows an example of the control panel window on the screen. This icon 229 of Fig. 19 appears when the digital VTR 203 is connected to the LAN 4. The digital VTR control panel object 221 depicts a window of Fig. 20 by default on the screen. In this window, there is an option menu 232 for selectively display the control panel windows on the screen. As the tape is running, the passed time is displayed in a time counter box 265. As the digital VTR 203 has a number of control modes,

there is a mode option box 267 containing a first switch button for setting a control mode by default and a second switch button 268 for selectively setting more elaborate control modes. A rewind button 269, a reverse play button 270, a pause button 271, a play button 272, a fast feed button 273, a stop button 274 and a record button 275 are displayed in array.

(0071)

Fig. 21 is a diagram, partly in pictorial form, to explain the correspondences between the classes the objects belong to and the constituent elements of the "control panel for digital VTR" object. All classes to which the fundamental constituent elements belong are defined previously in the class library 1081. This library is kept in the multimedia controller 1. As is obvious from Fig. 21, all the constituent elements of the "control panel for digital VTR" object 221 function as respective individual objects constituting the "control panel for digital VTR" object 221.

(0072)

In Fig. 21, the frame of a control panel window 231 on the screen corresponds to the VTR control panel object 221 (ID = 1) of panel class. The control panel display option menu 232 corresponds to the panel view setting menu object 285 (ID = 2) of menu class. The time counter box 265 corresponds to the time counter object 286 (ID = 3) of form class. The rewind button 269 corresponds to the rewind button object 287 (ID = 4) of

button class. The reverse play button 270 corresponds to the reverse play button object 288 (ID = 5) of button class. The pause button 271 corresponds to the pause button object 289 (ID = 6) of button class. The play button 272 corresponds to the play button object 290 (ID = 7) of button class. The fast feed button 273 corresponds to the fast feed button object 291 (ID = 8) of button class. The stop button 274 corresponds to the stop button object 292 (ID = 9) of button class. The record button 275 corresponds to the record button object 293 (ID = 10) of button class.

(0073)

The control mode dialog box 266 corresponds to the control mode selection object 294 (ID = 11) of button group class. The first switch button 267 corresponds to the default button object 295 (ID = 12) of radio button class. The second switch button 268 corresponds to the advanced button object 269 (ID = 13) of radio button class.

(0074)

Next, of the objects constituting the "control panel for digital VTR" object 221 shown in Fig. 21, an example of the play button is taken to explain how to make up an object 290 therefor.

(0075)

Fig. 22 is a diagram to explain the formation of the play button object 290. In Fig. 22, there are shown elements 297, 298, 299, 300, 601, 602, 603, 604,

605, 606, 607, 608, 609, 610 and 611 written in the "control panel object" section of the "deputy digital VTR object" file 210.

(0076)

A piece of information 297 for recognizing the object are formed with a class name 298, an object ID 299 and a superior object ID 300. A first piece of information 601 for drawing an article of the object are formed with data 602 for location and size, data 603 for pattern and color and an object image 604. A second piece of information 605 for drawing an article of the object are formed with data 606 for location and size, data 607 for pattern and color and an object image 608. Object link data 609 consist of a link terminal object ID 610 and an outgoing message 611.

(0077)

The play button object 290 of button class is generated by information from that class and the "'control panel of objects' object" section 247 in the "deputy digital VTR object" file 210. A portion 613 accommodates a pointer that goes to a class method table 625, where it points to methods in the button class. The button class method table 625 is formed from means 626 responsive to start of generation of an object of button class for initializing the internal variables of the button object are initialized, means 627 for depicting the button object to display, and click response means 628. The user moves the mouse or like pointing device

to position the cursor 230 on top of the play button. Responsive to clicking on this button, the click response means 628 changes the display of the button for a moment to inform the user of the fact that the button object has been activated and sends a message to another object.

(0078)

The definition of every data processing means in the method table for these button classes is described in each class. Therefore, not only the play button object 290 but also all the other objects which belong to the button class share the common button class method table. 614 is message communicating means; 615 is processing and retrieving means; 616 is a portion for methods; and 620 is a portion for internal data. The internal data are of an object ID 621, the state 622 of the button, drawing parameters 623 and link data 624. The one of the types of the internal data which not only the play button object 290 but also all the other button objects which belong to the button class should possess is described in the class.

(0079)

The system director object 205 first reads in the " deputy digital VTR object" file 210, when to make up any of the objects. In the example of Fig. 22, it then accesses the data for recognizing the object and, on the basis of the description of its class name 298, forms the objects of the button class. For the play button object 290 to generate, the system director object 205

cooperates with the button initializing means 626 to initialize the "internal data" portion 620. According to the example of Fig. 22, the object ID is set to ID = 7 by the description of the object ID 299. From the description of the superior object ID 300, the system director object 205 recognizes that the play button object 290 belongs to the "control panel for digital VTR" object 221. In such a manner, based on the information from the object to any of the objects of principal interest, the system director object 205 recognizes which one of the objects contains the other. Thus, a number of the constituent objects are put together to form a complex object.

(0080)

The button drawing means 627 depicts the play button object 290 on the basis of the drawing parameters 623 and the data 622 of the state of the button. The button drawing means 627 is automatically activated when the button object is generated and when the superior object moves.

(0081)

The first piece of information 601 describes that article 625 of the button which is effective in the situation when it is not pushed. The data 602 for location and size are used to determine a location at which the play button object 290 is depicted in the window for the "control panel for digital VTR" object 221 and a size to which a rectangular frame of the play button 625

is limited. To figure the rectangular frame, as the button is not pressed, the data for drawing the button article 625 are expressed in the coordinates of the "control panel for digital VTR" object 221. Suppose, for example, the left hand upper and right hand lower corners are taken into account, then the data of that rectangular frame have a form like (X1, Y1) and (X2, Y2). For the play button object, when not pressed, its pattern and color are determined based on information either from the pattern and color data 603 or from the object image 604. The data 603 for pattern and color are described in an appropriate language to draw lines and paint colors, that is, to describe the object in a graphical form. The object image 604 is expressed in the form of bit map data. In general, the expression by the former costs a less amount of data, but the latter has rather a high degree of freedom.

(0082)

The second piece of information 605, similarly to the first piece of information 601, describes that article 626 of the button which is effective in the situation when it is pressed. Based on both of the first data 601 and the second data 605, the values of the drawing parameters 623 are determined. The link data 624 are set in based on information from the data 609 for object linkage. Therefore, as the message to transmit, "play" is set in and, as the link terminal object ID, a link terminal object ID is set in. In connection with

the latter, it is to be noted that when to transmit the message, only one of the terminal objects over the entirety of the system should be selected to receive this message. To this purpose, the link terminal object ID to be used is set in the preceded form by the device ID the system director object 205 has assigned to the digital VTR, when the digital VTR 203 was connected to the LAN 4.

(0083)

Even if it happens that two devices have their link terminal objects to use the same ID, therefore, the message can be transmitted right to the desired object. The button state data holds the information of whether or not the button is pressed.

(0084)

Fig. 23 is two flowcharts, one of which shows the operation when the user positions the cursor 229 on top of the icon 229 of the digital VTR 203 and double clicks, and another one which shows the operation when the user has manipulated the control panel.

(0085)

Fig. 24 is a plan view of the display of a window for the multimedia controller 1 on the screen as presented when the user has double clicked on the icon 229 of the digital VTR. In Fig. 24, the control panel window 231 for the VTR 203 is selected by default, and the play button is shown at 272.

(0086)

Fig. 29 is a diagram of the relationship between the structure of the "control panel for digital VTR" object of panel class and the object description data.

(0087)

In Fig. 29, a portion 1401 accommodates a pointer that goes to the class method table, in this instance, a panel class method table 1402. This table is formed with panel initializing means 1403 for initializing the panel object, panel drawing means 1404 for showing the panel in a graphical form, and click response means 1405 for activating the clicked object. Message communicating means 1406, processing and retrieving means 1407, and an "internal data" part 1410 are shown. The internal data are of an object ID 1411, the panel state 1412, and drawing parameters 1413. The "internal data" part 1410 is initialized according to the description of the "deputy digital VTR object" file 210. The "'control panel for digital VTR' object" section 211 in this file 210 comprises data 1414 for recognizing the object, a first pack 1418 of data for drawing an icon 1426 of the digital VTR 203, and a second pack 1422 of data for drawing the frame of the control panel window for the digital VTR. The data 1414 for recognizing the object are of a class name 1415 (panel class), an object ID 1416 (ID = 1), and a superior object ID 2417. The first data pack 1418 consists of data 1419 for location and size, data 1420 for pattern and color, and an object image 1421. The second data pack 1422

consists of data 1423 for location and size, data 1424 for pattern and color and an object image 1425.

(0088)

Referring to Fig. 23 and Fig. 29, the routines for displaying the control panel window for the digital VTR 203 and activating the play mode are described below. As described in connection with the routine of Fig. 16, at a time when the system director object 205 has generated the deputy digital VTR object 220, the deputy digital VTR object 220 presents the display of the icon 229 as obtained based on the icon image 1426. For now, when the user double clicks on the icon 229 for the digital VTR (643), the control panel object 221 of the deputy digital VTR object 22 sends a message of executing the drawing function to all objects constituting the control panel object 221. Based on this message, all the objects shown in Fig. 21 activate the drawing means. Meanwhile, the control panel object 221 depicts the frame of the control panel window for the digital VTR based on the second pack of object drawing data. As a result, the digital VTR control panel window 231 is displayed on the screen (644) as shown in Fig. 24 and waits for instructions from the user (645). With this, when the user positions the cursor 230 on top of the play button 272 and clicks (646), the control panel object 221 sends a message "PLAY" to the controller object 214 of the digital VTR 203 (647). Responsive to this message, the controller object 214 of the digital VTR 203 activates

the play executing means (648), thus starting an operation of the play mode of the digital VTR 203.

(0089)

As has been described above, according to the invention, when a multimedia device is only connected to the multimedia controller via the LAN, its object necessary to coordinate the multimedia device with the others, or deputy multimedia device object, is automatically generated in the multimedia controller. Further, the control panel necessary to choose the multimedia devices is automatically displayed in the multimedia controller window on the screen. With the help of this control panel, the user activates an item. Then an unique message is transmitted to the controller object of the corresponding multimedia device. So the desired functions are executed. Since the information necessary to generate the deputy multimedia object, which in turn is necessary to manipulate the multimedia device, is obtained from the "deputy multimedia device object" written file as read from the multimedia device, what suffices for the multimedia controller is only the fundamental class library. So there is no need to store the related database to any specific multimedia device in advance.

(0090)

Fig. 25 is a diagram of the relationship between the structure of the deputy "data input to digital VTR" object and the data for describing the object. In

Fig. 25, the deputy "data input to digital VTR" object 222 contains a portion 668 for accommodating a pointer that goes to the class method table, in this instance, a deputy data input class method table 679. This table is formed with means 680 for initializing the deputy data input object, means 681 for updating the link data and compatible file type reply means 678.

(0091)

669 is message communicating means; 670 is processing and retrieving means; 671 is a "method" part; and 674 is an "internal data" part. The internal data are of an object ID 675, another ID 676 which represents the related data input object, compatible file types 677 and links 1006 with data output objects.

(0092)

The "deputy digital VTR object" file 210 contains a "deputy 'data input/output of digital VTR' object" section 212. Based on the information from this section, the deputy "data input to digital VTR" object is generated. The data for the "deputy input" object described in the section 212 are of an object ID 683 (in this instance, ID = 1), a related data input object ID (in this instance ID = 1), and a list 685 of compatible file types (in this instance, assumed to be formats so called "AV1" and "AV2"). According to the description of these parameters, the deputy input object initializing means 680 initializes the "internal data" portion 674.

(0093)

Fig. 26 is a diagram of the relationship between the structure of the deputy "data output from digital VTR" object and the data for describing the object. In Fig. 26, the deputy "data output from digital VTR" object 223 contains a portion 690 for accommodating a pointer that goes to the class method table, in this instance, a deputy data output class method table 1048. This table is formed with means 694 for initializing the deputy data output object, means 695 for updating the link data and compatible file type reply means 700.

(0094)

691 is message communicating means; 692 is processing and retrieving means; 693 is a "method" part; and 696 is an "internal data" part. The internal data are of an object ID 697, another ID 698 which represents the related data output object, compatible file types 699 and links 688 with the data output object.

(0095)

Based on the information from the "deputy digital VTR object" file 210 at the "deputy 'data output from digital VTR' object" section, the deputy "data output from digital VTR" object is generated. For now, the data 1001 for the deputy data output object described in the section 212 are of an object ID 1002 (in this instance, ID = 1), a related data output object ID 1003 (in this instance ID = 1) and a list 1004 of compatible file types (in this instance, assumed to be formats so called "AV1" and "AV2"). According to the description

of these parameters, the deputy data output object initializing means 694 initializes the "internal data" portion 696.

(0096)

Fig. 27 is a diagram of the structure of the "data input to digital VTR" object. This object 208 includes a portion 1030 for accommodating a pointer that goes to the class method table, in this instance, a data input class method table 1031. This table 1031 is formed with file writing means 1032, data receiving means 1033 and link data updating means 686. 1023 is message communicating means; 1024 is processing and retrieving means; 1025 is a "method" part; 1028 is an "internal data" part; 1029 is an object ID; and 1030 is link data.

(0097)

Fig. 28 is a diagram of the structure of the "data output from digital VTR" object. This object 209 contains a portion 1035 for accommodating a pointer that goes to the class method table, in this instance, a data output class method table 1044. This table is formed with file reading means 1045, data transmitting means 1046, and link data updating means 687. 1036 is message communicating means; 1037 is processing and retrieving means; 1038 is a "method" part; 1041 is an "internal data" part; 1042 is an object ID; and 1043 is link data.

(0098)

After the deputy data input object 222 and

deputy data output object 223 of the digital VTR 203 have been generated in the multimedia controller, these objects function as if they were chief ones, or the "data input to digital VTR" object 208 and the "data output from digital VTR" object 209. Now suppose the digital VTR receives, for example, a file from another multimedia device by the copy function, then the system director object 205 inquires of the deputy "data input to digital VTR" object 222 what types of files are possible to input. Responsive to this inquiry, the compatible file type reply means of the deputy "data input to digital VTR" object 222 gives off information about the file types the digital VTR 203 can accept.

(0099)

If the type of the file to be copied is found to be present among them, a link is established from the deputy output object of that multimedia device which has the file to be copied to the deputy "data input to digital VTR" object 222. The link data updating means 681 of this object 222 sends a message to the "data input to digital VTR" object 208. As the link updating means 686 of this object 208 is activated, the link data of the "data input to digital VTR" object 208 are updated.

(0100)

At the same time, the deputy data output object of that multimedia device which has the file to be copied sends a message for updating the link data of the data output object. As the link data are updated, a link is

established from the data output object of the multimedia device which has the file to be copied to the "data input to digital VTR" object 208.

(0101)

After this, the data transmitting means of the data output object of the multimedia device which has the file to be copied is activated. The data output object of the multimedia device which has the file to be copied sends a message to the "data input to digital VTR" object. As the data receiving means 1033 and the file writing means 1032 are activated, copying of the file is carried out. In short, when the user gives the copy command or any of the other commands to the deputy data input object and the deputy data output object in the multimedia controller, the deputy data input object and the deputy data output object send the messages to the data input object and the data output object of the main systems of the respective multimedia devices. Thus a link for data communication is established between these two multimedia devices. Concerning the operation of copying data, for example, it is not virtually necessary for the multimedia controller to take direct participation.

(0102)

As is understandable from the foregoing, according to the prior art, when to make control of the entirety of a system having a plurality of multimedia devices connected to one another, the device drivers or

like applications for this purpose had to be previously installed in the controller. According to the invention, however, such a necessity is obviated. So, when a multimedia device is only connected to the LAN, the control panel and the status of the network are automatically displayed in the controller window on the screen. With the help of the windows on the screen, therefore, it becomes easy to turn on and off the power sources of the devices, control the main systems of the devices, and coordinate inputting and outputting of the various signals and data.

(0103)

Another advantage is that of the items the multimedia devices have sent to the control panel of the controller, the ones which are identical in definition to the items the controller has already possessed may be exchanged either in part or all therebetween, depending on the user's taste. As the user interface differs with different makers, it is thus made possible to unitarily rearrange the items.

(0104)

Yet another advantage is that it becomes possible to execute the control functions from a controller in the distant place and the functions of accessing to the terminal multimedia devices in a transparent fashion over the LAN.

(0105)

In the following, let us describe a function of regulating message flow in a system where a number of multimedia devices are connected to a controller via a LAN such as that described above. As a specific example, suppose the user wants to copy the data of a digital camera to a digital VTR. The digital camera of this example is assumed to be of the handy type so that it can be used outdoors to record motion pictures and sound and to have communicating means to be used with the system described above.

(0106)

Fig. 30(a) shows control panels displayed at once on a screen of the display device of the multimedia controller described above. The control panel 301 is used for controlling the digital camera device and the control panel 310 is used for controlling the digital VTR device. To generate the control panels, the system director accesses these devices and therefrom reads the respective individual file sections 1062 (Fig. 7) for describing the "control panel for multimedia device" object into the multimedia controller. After they appear on the display of the multimedia controller, a message switching network waits for an event from the user.

(0107)

In Fig. 30(a), boxes 302 and 311 represent the tape cassette chambers, indicating whether or not the devices have been loaded with the respective tape

cassettes. Boxes 303 and 312 represent counters for measuring the time for which the tape has run. Button objects 304 and 313 each display a list of contents of the tape used in the respective one of the devices. When the mouse clicks on this button object, a list of current contents of the tape is displayed. 305 and 314 are Play button objects. By clicking on this button object, the corresponding device is started reproducing.

(0108)

306 and 315 are FF button objects. When the mouse clicks on this FF button object, the corresponding device starts fast feeding the tape. The tape is rewound by clicking a RWD button object 307, 316. As the tape is running, when the user clicks on a Stop button 308, 317, the corresponding device stops the Play, FF, RWD or Rec function from further operating. 309 and 318 are Rec button objects. When the mouse clicks on this button object, the corresponding device starts recording the video or audio data as transmitted from the outside over the LAN or like communication path.

(0109)

Using Fig. 30, as the user manipulates such a user interface to dub video and audio data from the digital camera 301 to the digital VTR 310, the message flow is described.

(0110)

The user acts on the user interface at a time of Fig. 30(a). From within the control panel 301 for

the digital camera, the user starts a dragging operation. At this time, a black oversize frame 319 appears as shown on the left hand half of Fig. 30(b). With the control panel 301 remaining stationary relative to the screen, the user goes on dragging. At a point in time when the mouse cursor crosses the right hand side of the black frame, an arrow 321 appears with its head in coincidence with the mouse cursor. As the mouse continues moving, the mouse cursor eventually enters the interior of the frame of the control panel 310 for the digital VTR. At this moment, a black oversize frame appears. Here the user ends the dragging operation, leaving the display of a linkage between the objects with a direction from 301 to 310 being presented.

(0111)

At this point in time, it is in the interior of the the multimedia controller that determination of whether such a linkage is valid or invalid is started to make. If the linkage is found to be invalid, the display returns to the state of Fig. 30(a). If the linkage is found to be valid, the display is left unchanged from the state of Fig. 30(b).

(0112)

In the following, as the user has selected a link, the manner in which the message flow is regulated by the multimedia controller is described by reference to Fig. 31, Fig. 32 and Fig. 33.

(0113)

Fig. 31 shows the structures of objects within the multimedia controller 322 and the digital VTR device 338 and the digital camera device 339 and the status of current connections of these devices. The objects in each of the devices are all able to switch messages, for data are transferred from one to another and vice versa. For every one of the devices, it is also possible to transmit or receive messages and data to or from any of the objects in the other devices by the respective communicating means 336, 406 and 407 over the LAN. All of the objects of Fig. 31, therefore, can perform the functions of switching messages and transferring data from any one of the objects to any one of the others over the communication paths.

(0114)

The adjacent two of object levels shown in Fig. 31 are expressed by the superior object ID in the fundamental structure of the objects described above. As mentioned before, 323 is the system director object. The system director object 323 has its "methods" storage part made to include means responsive to anticipation of inputting or outputting data from or to another device for checking the input or output data to see if the formats of the source and destination devices are compatible with each other. This means is shown at 324 named "input/outout data checking." Another storage part 335 contains objects that act as substitute for the respective connected devices. To generate such a deputy

object for each connected device, the initializing means therefor described before reads that item of information which is necessary for initialization from the respective one of the devices connected to the LAN 337, when the system is powered on, or at the termination of each predetermined period.

(0115)

In the present embodiment, the digital VTR type device 338, the digital camera type device 339 and other devices (not shown in Fig. 31) are connected to the LAN 337. It is assumed that the information for initializing is read from these devices, when the deputy "digital VTR" object 326, the deputy "digital camera" object 331 and the other deputy objects for the devices not shown in Fig. 31 are to form in the multimedia controller. A deputy "data input to digital VTR" object 325 has response means to an incoming message for inquiring the digital VTR type device about the input data. A deputy "data output from digital VTR" object 330 has response means to an incoming message for inquiring the digital VTR type device about the output data. A "controller of digital VTR type device" object 340 controls the hardware of the digital VTR type device in accordance with the message from the other object. A "controller of digital camera type device" object 341 controls the hardware of the digital camera type device in accordance with the message from the other object. An "input to digital VTR type device" object 408 has data

receiving means by which it can receive the data as are being transferred to the destination address which coincides with its own object ID. An "output from digital camera" object 409 has data transmitting means. This means adds information of the destination address ID or the like to the output data from a reproduction apparatus (not shown) in the digital camera, and then produces the result at the output of communicating means 407, from which to transmit the data. The communicating means 336, 406 and 407 thus allow all the connected devices to communicate with one another over the LAN.

(0116)

Turning back to Fig. 11, there is shown the structure of components of the system director object. When the user selects a link between two of the devices, the Window Server sends a message to the processing and retrieving means 342. Responsive to this, the means 342 activates the data input/output checking means 343. The checking means 343 performs a function of testing the data to see if the formats in the linked two of the devices are compatible with each other.

(0117)

Fig. 32 is a flowchart of the routine for the operation of the data input/output checking means. With the help of the user interface such as that of Fig. 30, when the user selects an outgoing link from a device A to a device B, the Window Server sends a message representing that the outgoing link from the device A

to the device B is set in, or Linked Message to the system director object 205 (Fig. 11). On receipt of this message, the processing and retrieving means 342 (Fig. 11) accesses the "methods" storage part and activates the input/output data checking means.

(0118)

In the following, for the case that the user has chosen the devices A and B to link up, the manner in which the input/output data checking means 343 operates is described using the flowchart of Fig. 32.

(0119)

At first, in S1, the checking means 343 inquires of the deputy "output from device A" object what file type it can accept. (If there are two or more acceptable file types, their list is replied).

(0120)

Then, S2 inquires of the deputy "input to device B" object what file type (format) it can accept. If there are two or more acceptable file types (formats), it even replies information about the order of priority of all the file types at the same time. It is to be noted here that the priority order is determined by either the user or the manufacturer of the device B, depending on which one of the file types is currently more suitable to the device B than the others. The highest file type in this priority order is called the "most prior" file type. Given the priority order of the device B, the file type (or the list of acceptable file types) in the

device A is inspected (or traced) in S3. That is, the checking means 343 searches for the most prior file type for the device B among the available ones the device A has as read in the step S1. Let us call the thus obtained file type "most compatible."

(0121)

S4 examines whether or not the S3 has failed to find it out. If the acceptable file type list of the device A and the acceptable file type list of the device B have no common file, then branch to S8. If succeeded, then advance to S5.

(0122)

The S5 examines whether or not the data attribute of the most compatible file type is coincident with that of the most prior file type. If so, then advance to S6.

(0123)

As the link in question is determined in the S5 to be valid, the S6 accesses the "internal data" storage part 1076 (Fig. 11) in the system director object and selects the current link tracking data 344 (Fig. 11) to save the information representing that a new valid link has been set in with its direction from the device A to the device B and the information of the data attribute and the file type useful in this link.

(0124)

S7 informs the "Linked Message" source object, the deputy "output from device A" object and the deputy

"input to device B" object of the fact that a new valid outgoing link from the device A to the device B has been set in and of the file type useful in this link. (The deputy input object and the output object of each of the these devices must be updated to reflect such changes in the status of the network.) A normal sequence of steps thus finishes.

(0125)

If no compatible file type is found in the S4, then advance to S8, where the link in question is determined to be invalid, informing the "Linked Message" source object of the fact that, because of no common format being available, the preliminary link has been rejected. S11 requests the "Linked Message" source object for erasing the corresponding entry to that link. Thus, the processing ends.

(0126)

Again, when, in the S5, the most compatible file type and the most prior file type do not match each other in respect to the data attribute, S9 sends a message for demanding a warning display to present the "Linked Message" source object. The content of this warning is to inquire of the user whether or not to satisfy if the attribute of the most compatible file type is to use for transferring data. So an event loop is started to wait for the user to make decision of whether to continue setting of the link in question or cancel the preliminary link. If, in S10, the user has commanded continuing,

then advance to the S6. If canceled, then return to S11 and send a message for erasing the corresponding entry to the preliminary link. After this, the procedure finishes.

(0127)

In the following, using Fig. 30, Fig. 11 and Fig. 33, for the case that the user has selected an outgoing link from the control panel 301 (Fig. 30(a)) for the digital camera to the control panel 310 (Fig. 30(a)) for the digital VTR, the manner in which the multimedia controller operates is described in more detail below.

(0128)

On the user interface such as that of Fig. 30, the user achieves a linkage 321 (in Fig. 30(b)) with the direction from the control panel for the digital camera to the control panel for the digital VTR. Responsive to this, the window server sends a message informing of the fact that the digital camera and the digital VTR have been selected to link up to the system director object of Fig. 11.

(0129)

On receipt of this message, the processing and retrieving means 342 in the system director object selects the input/output data checking means 348 from the "methods" storage part to operate. The input/output data checking means 343 performs the steps like those of the flowchart of Fig. 33 to determine whether the link

between the control panel for the digital camera and the control panel for the digital VTR is valid or invalid.

(0130)

At first, S1 inquires of the deputy "output from digital camera" what file type is available. A table shown in Fig. 45(a) is of available file types from the digital camera in the present embodiment. Because there are many available file types, they are shown in the form of a list. The available file types are, as shown in Table 1, combined with the data attributes In here, Audio represents the attribute for audio data, Movie for motion picture data, and Audio & Movie for synchronized audio and video data.

(0131)

Then, S2 inquires of the deputy "input to digital VTR" object what file type it can accept. In the present embodiment, the digital VTR has a plurality of acceptable file types as shown in a table of Fig. 45(b). All the file types are let to be known together with the priority order. In the present embodiment, as shown in the Table 2, the file type AM4 with the data attribute of Audio & Movie is given No. 1, meaning the most prior format for the VTR.

(0132)

Then, S3 traces the table of Fig. 45(a) and, according to the priority order of Fig. 45(b), searches for the one of the file types which falls in coincidence. From the table of Fig. 45(b), the AM4 is found to be at

a first place in the priority order. Therefore, while tracing the table of Fig. 45(a), the S3 searches for the AM4 format. Because there is no AM4 format in the table of Fig. 45(a), the AM5, because of being at the 2nd place in the priority order, is then searched for in the table of Fig. 45(a).

(0133)

If this, too, has failed, such a searching procedure repeats itself in succession from the upper to the lower places in the priority order to inspect the file type of coincidence. On reaching to the 4th place in the priority order, a file type of Movie 2 is found to coincide for the first time ever. This is taken as the most compatible file type for the link from the digital camera to the digital VTR.

(0134)

Since the searching in the S3 has succeeded in finding out the coincident file type, the process goes from the S4 to an S5.

(0135)

In the S5, since the Movie 2 found as the most compatible file type in the S3 has the data attribute: Movie, because this does not coincide with the data attribute of the most prior file type, that is, the Audio & Movie, the process goes to an S9.

(0136)

The S9 sends a message for demanding a warning display to present as shown in Fig. 34 to the window

server and enters a loop S12 that waits for a command from the user.

(0137)

In the present embodiment, the user is assumed to click on a "YES" button in the panel of Fig. 34 in an S10. The S10 is followed by an S6.

(0138)

The S6 accesses the "internal data" storage part in the system director object and saves the information of the fact that a valid outgoing link is set in from the digital camera to the digital VTR by the file format: Movie 2 with the data attribute: Movie as part of the current link tracking data 344 (Fig. 11).

(0139)

In final, S7 informs the deputy "output from digital camera" object and the deputy "input to digital VTR" object by messaging of the fact that the valid link with the data attribute of Movie 2 has been set in from the digital camera to the digital VTR. After this, the sequence of steps of the input/output data checking means finishes.

(0140)

In the S7, on receipt of the message from the input/output data checking means, the deputy "output from digital camera " object 410 (Fig. 41) activates updating means 413 (Fig. 41) for the link entries. The updating means 413 then starts store the information of the link with the digital VTR with the format of Movie 2 in an

entry of data 419 (Fig. 41) in the "internal data" storage part and at the same time informs the corresponding data input object ID, that is, the data output object 420 (Fig. 42) of the digital camera of the fact that an event of updating the link information has occurred and also of its content. On receipt of this, the data output object 420 (Fig. 42) of the digital camera activates updating means 425 (Fig. 42) for the link data to save the updated content of the link in the link data 428 (Fig. 42).

(0141)

The deputy "output from digital camera" object 410 (Fig. 41) further sends a message to all the objects which belong to the control panel object 333 (Fig. 31) for the digital camera. This message commands that, of these objects, the ones which do not partake in outputting data be grayed out to reject the access from the user. So with the record button 309 of Fig. 30(b) grayed out, when the user clicks on it by the mouse, the record mode is hindered from operating.

(0142)

The deputy "input to digital VTR" object 222 (Fig. 25), too, receptive of the message from the system director object activates the updating means 681 for the link entries. This means 681 starts storing the information of the fact that an incoming link has been set in from the digital camera in the format of Movie 2 as the data 1006 in the "internal data" storage part

and informs the corresponding data input object ID or the data input object 686 (Fig. 27) of the digital VTR of the updated contents of the link entries. On receipt of this, the data input object 208 (Fig. 27) of the digital VTR activates the updating means 686 (Fig. 27) for the link entries to save the updated contents of the link entries in the link data 1030 (Fig. 27). The deputy "input to digital VTR" object 222 (Fig. 25) further sends a message to all the objects which belong to the control panel object 328 (Fig. 31) for the digital VTR. This message commands that the ones of the objects which do not partake in inputting data be grayed out to reject what the user inputs.

(0143)

By this message, the Play button 314, the FF button 315 and the RWD button 316 are grayed out as shown in Fig. 30(b). So, even if the user clicks on any of these button, no effect results. Under this condition, the link is maintained valid as shown in Fig. 30(b).

(0144)

After the valid link has been set in as shown in Fig. 30(b), the internal data takes values as shown in Fig. 44. In Fig. 44, (a) is the internal data of the deputy "output from digital camera" object; (b) those of the deputy "input to digital VTR" object; (c) those of the "data output from digital camera" object; (d) those of the "data input to digital VTR" object; and (e) those of the system director object. The updated internal

parameters by setting up the link by the user are shown in square brackets.

(0145)

With the valid link set in as shown in Fig. 30(b) and the internal data taking the values shown in Fig. 44, when the user clicks on the play button 305, the digital camera 301 starts transferring the data of the Movie 2 format to the digital VTR.

(0146)

Further, when the user clicks on the record (Rec) button 318, the digital VTR starts recording the data as transmitted to itself from the digital camera 301. Hence, dubbing takes place between the two multimedia devices. At times during this operation, the multimedia controller transmits and receives to and from these devices as follows:

(0147)

The user first clicks on the play button 305 (Fig. 30). The window server determines the values of coordinates (in the control panel) of a point at which the mouse cursor lies when clicked and transfers the result to the control panel object 333 (Fig. 31). On comparison with the internal data, the values of coordinates of this point are found to coincide with those for the play button. Since this is interpreted as having pushed the play button, the control panel object 333 transmits information representing that the user has given "data output" command (or a "play" message) to the

controller object 341 in the digital camera over the communication line. In the digital camera controller object 429 (Fig. 43), the processing and retrieving means 431 activates the corresponding one of the means in the "methods" storage part to this message, in this instance, the play executing means 433. The play executing means 433, on one hand, controls the hardware of the digital camera to thereby render the digital camera operative. That is, part of the mechanical system is made ready to operate in the play mode. On the other hand, it requests the "data output from digital camera" object 409 (Fig. 31) for producing data. Responsive to this message, the processing and retrieving means 421 (Fig. 42) in the object 409 selects the data reading means 423 and the data transmitting means 424 from the "method" part and activates them to operate. The data reading means thus starts reading information from the recording medium.

(0148)

The data transmitting means, too, starts looking up the link information 428. Since, at this time, the content of the link information 428 is such that, as shown at 430 of Fig. 44, the data destination object ID = 120 and the file type = Movie 2, the data transmitting means changes the type of read data file to Movie 2 and then transmits the data to the object of ID = 120. In this example, the data of Movie 2 are motion picture. As shown in Fig. 36, at the top of the data, there is a header

consisting of the file type, the numbers of vertical and horizontal pixels and others such as information necessary to reproduce the motion picture. The message text is divided into packets of data with the boundaries marked by time data called "time stamps." Editing can be done in units of this time stamp.

(0149)

ID=120 selects the "data input to digital VTR" object 208 (Fig. 27) to receive the data of Movie 2 from the object of ID=100. The object 208 looks up its own link data 1030 (Fig. 27). Since, as shown at 431 in Fig. 44, there is the link with the data output object of ID=110 in the Movie 2 file type, the object 208 then stores the header 364 (Fig. 36) for the data to be later transmitted in the memory 370 (Fig. 37).

(0150)

The user eventually clicks on the record button 318 (Fig. 30) in the control panel for the digital VTR. The window server determines the values of coordinates (in the control panel) of a clicked point and transfers the result to the control panel object 328 (Fig. 31). On receipt of this, the object 328 looks up its own internal data. Since, as these values of coordinates fall in the range for the record button, this is interpreted as having pushed the record button, the object 328 transmits information representing that the user has given a "data record" command to the controller object 341 in the digital VTR over the

communication line. On receipt of this, the processing and retrieving means 1011 (Fig. 15) in the controller object 341 selects the record mode setting means 1020 (Fig. 15) to be activated. The setting means 1020 controls the hardware of the digital VTR to thereby render the mechanical system in an operating state so that the record mode is ready to operate. After this, the setting means 1020 requests the "data input to digital VTR" object 208 (Fig. 27) for recording the inputted data. Responsive to this request message, the object 208 controls the hardware of the digital VTR to thereby read information from the header 364 (Fig. 36) stored in the RAM 376 (Fig. 37). After the header, the text is transmitted packet by packet with the time stamp at the lead. So, the "data input to digital VTR" object 208 starts recording the packets of data in continuous sequence on the magnetic recording medium 374. The thus recorded motion picture data of Movie 2 contain the header information necessary to reproduce the motion picture. Therefore, it is possible to do reproduction.

(0151)

(Second Embodiment)

A second embodiment of the invention is next described. The multimedia controller in this embodiment has the same features as those in the first embodiment and an additional feature that there is an environment for the operation of application software (utility software).

(0152)

In the following, the multimedia controller in the present embodiment is described. (In the invention, the application software and the utility software do not substantially differ from each other. So the utility software also is hereinafter called the "application" software.)

(0153)

Fig. 39 shows the structure of objects in the interior of the multimedia controller in the second embodiment of the invention. The multimedia controller in the second embodiment of the invention includes means for installing an application (utility) for the multimedia controller and has the environment in which the application operates.

(0154)

In the multimedia controller of the present embodiment, the application is installed from the floppy disk or communication means to an storage 388 (Fig. 39) for the application class in the multimedia controller. The content of the application in the storage 388 is a file for describing a class based on the object oriented concepts. Using this class, the application object forming means 386 (Fig. 39) produces an object that is able to execute the application in a work area 389 (Fig. 39). Thus, the application is rendered operative. The application object depicts a window object in graphically visual form on the screen of the display for

the multimedia controller. Using the mouse or like pointing means, the user selectively activates the items on the window object. Responsive to this, the window server sends a message of how manipulation is done at what coordinates to the application object. Since the application object has its own internal data including those about what graphics are present and where they are located, it can recognize which icon is clicked and which one of the icons is dragged to another by using the information from the window server.

(0155)

As one specific example of software, the connection constructor application is described below.

(0156)

Fig. 38 shows a user interface for the connection constructor. In Fig. 38, a connector constructor window 370 contains a cursor of the mouse as the input means of the user and icons 379 to 383 for a CD-player, a VTR, display, a CA-TV decoder, a DAT deck, an amplifier receptive of audio signals for producing voices or sound from a speaker (not shown), an MO-player for recording on or reproducing from a magneto-optical disk, a multimedia controller and an LD-player respectively. When the devices are selectively put to correlation (connection) for transmitting or receiving data therebetween, the paired relations are indicated by the respective arrows. With the help of these arrows, the user can see all current links with their directions

at a glance.

(0157)

In this example, the LD-player 383 (Fig. 38) is shown with no connection to anything whatever. But, even under such a condition, the LD-player may be rendered operative in the reproduction mode. If so, it is linked up with the multimedia controller by default. The video and audio data are then transferred to that object in the multimedia controller which governs displaying of a picture and production of voices or sound.

(0158)

In such a manner, the connection constructor window, when opened on the screen, is able not only to display the current connections, but also to assist the user in editing the connections. The user can establish a new connection by dragging one of the icons to another. In Fig. 38, there is shown the middle course of connecting the CD player 371 to the amplifier 367 by the user dragging the mouse.

(0159)

The user clicks on the one of the icons on the screen of the multimedia controller display 27 (Fig. 4) which represents the connection constructor, (or the object of the connection constructor). The window server sends a message informing of the fact that the icon has been selected to the connection constructor object 390 (Fig. 39). On receipt of this message, the connection constructor object 390 cooperates with the

processing and retrieving means (not shown) to activate means 391 (Fig. 39) for displaying the connection constructor window.

(0160)

Using a flowchart of Fig. 40, the manner in which, after the user has clicked on the icon for the connection constructor, messaging is transferred across the connection constructor object 364 from one of the objects to another and vice versa in the multimedia controller, is described below.

(0161)

The display means 391 (Fig. 39) for the connection constructor window operates in a first step S1 to request the system director object 205 (Fig. 11) for sending a list of the object IDs of all current devices connected to the network and then stands by in a step S2 till the data are sent in response to that message. The system director object looks up its own internal data and, by reference to the registered object data 1078 (Fig. 11), makes up a list of registered device objects and sends it back. This list is stored in the "internal data" part. Then, S3 follows.

(0162)

The S3 selects the ones of the objects IDs stored in the internal data which are identified with those of the connected devices in the list and sends all of them a message for demanding the graphics data for the respective icon to transfer. S4 waits for this data

to come. The graphics data, after having been received from all the object IDs, are stored in the "internal data" part. In S5, the discrimination object for the format of input/output data is requested for sending information about the links between any two of the devices, say which object ID is linked with which object ID in what data attribute. On receipt of the information of the current links in S6, as this means that the connection constructor object has got the information satisfactory enough to display a connection constructor window such as that 370 of Fig. 38, S7 depicts the connection constructor window. Before the depicting is started, the locations at which the icons are to appear are so determined by computation that the links do not intersect one another as far as possible.

(0163)

In the present embodiment, as shown in Fig. 38, the current links between the devices are expressed by the straight lines connecting the icons of the devices to each other. Thus, the user can quickly and easily perceive the status of the network. (Nonetheless, this expression is not necessarily given by straight lines and may be modified by using curved lines or the like, of course, provided all the connections get more comfortable to see.) It is also to be noted that, as the links differ with different data attributes, this differentiation is made by using different kinds of lines (or colors)

(0164)

In the present embodiment, as shown in Fig. 38, the visual type of data is indicated by the solid line, the audio type of data by the dashed line, the application program by the dot-and-dash line, and the text type of data by the double dot-and-single dash line.

(0165)

For example, in Fig. 38, the icon 372 of the VTR is connected to the icon 373 of the display device by a solid line arrow 379. This means that, if the VTR is actuated to operate, the display device is automatically selected as the destination to which the video data the VTR is to reproduce is transmitted over the communication line. In another case that the user wants to view the CA-TV, all what is needed to do is only open the control panel for the CA-TV and choose a channel. For the video data of the CA-TV, the link to the destination of the corresponding ID, that is, the display device 373, is then automatically selected to transmit over the communication line. Also for the audio data, the line to the destination of the corresponding ID, that is, the amplifier 376, is automatically selected to transmit over the communication line.

(0166)

It is also to be noted in the present embodiment that the data from the MO-player are of the program form. The program data are transferred to the multimedia controller wherein, as the header message has its data

attribute written as "program", the text message is recognized as the application program. So, this program is automatically activated. Such a function can be realized in the other devices as well as the multimedia controller. For example, the display device is provided with an operating system as arranged on receipt of a program to execute this program.

(0167)

It is in the present embodiment, that this program has video data buried in the interior thereof. When this program is being executed, therefore, pictures are reproduced on the display device. A merit of this method is that, because the video data are buried in the program, the necessity of checking the format of data between any two of the devices can be obviated.

(0168)

On such a window, the user can link one of the icons up with another by the mouse or like input means, thus editing the relationships for transmitting and receiving data between the paired devices. When a new connection is established between two of the icons, the connection editing means 394 sends a linked message to the system director object. On receipt of this message, the system director object starts to operate in a similar manner to that in the first embodiment as according to the flowchart of Fig. 33. In a case when the link is determined to be effective by the input/output data checking means in the system director object, the

one of the internal data of the object which is concerned with the link is updated likewise as in the first embodiment.

(0169)

Also, the connection constructor object is informed if the data attribute of the new link from the system director object. The connection constructor object selects the one of the kinds of lines which matches the data attribute and displays it. Thus, the internal data of that object which partakes in altering the link is made to conform with the display on the connection constructor window at all times.

(0170)

(Advantages of the Invention)

As is understandable from the foregoing, according to the invention applied to control all of a plurality of multimedia devices which are connected to one another to form a system, the system control method herein used has obviated the necessity of making an elaborate preparation by installing a device driver or like application as part of the software for every one of the devices in the controller as has been common up to now. The user needs only to connect the multimedia devices on the LAN, thereupon automatically causing the control panels for them and the status of the network to appear on the screen of the display of the controller. With these, the user can turn on or off the power sources, control the management of the main body of each of the

devices and select inputs or outputs, for example. Further, with the help of a user interface of simple form, the user can construct the relationships for transmitting and receiving data between the multimedia devices. Furthermore, when to transmit or receive data, it is the multimedia controller that checks the file format. Adjustment of the compatibility is thus automatically made. Therefore, the user no longer needs to do troublesome tasks such as that of checking every one of the available file formats.

(Brief Description of the Drawings)

(Fig. 1)

A diagram showing one form of the logic network of a multimedia controller and multimedia devices.

(Fig. 2)

Diagrams showing physical configurations of connecting a multimedia controller to multimedia devices.

(Fig. 3)

A diagram of the internal structure of an object embedded multimedia device.

(Fig. 4)

A diagram of the internal structure of an object embedded multimedia controller.

(Fig. 5)

A diagram of a hierarchy of systems in the multimedia controller.

(Fig. 6)

A diagram, of a hierarchy of systems in the multimedia device.

(Fig. 7)

A diagram of a multimedia controller and a multimedia device before they are connected to each other.

(Fig. 8)

A diagram of the multimedia device connected to a LAN.

(Fig. 9)

A diagram of the structure of a common class library.

(Fig. 10)

A diagram of the structure of an object.

(Fig. 11)

A diagram of the structure of a system director object.

(Fig. 12)

A diagram of the structure of a control panel section in a file for describing a deputy object.

(Fig. 13)

A diagram of the structure of a data input/output object section in the file for describing a deputy object.

(Fig. 14)

A diagram of an object embedded digital VTR before connected to the multimedia controller.

(Fig. 15)

A diagram of the structure of a VTR controller.

object.

(Fig. 16)

A flowchart of the operation that follows the connection of the digital VTR to the LAN.

(Fig. 17)

A plan view of a multimedia controller window on the screen.

(Fig. 18)

A diagram of the object embedded digital VTR connected as a multimedia device to the LAN.

(Fig. 19)

A plan view of an icon for the digital VTR.

(Fig. 20)

A plan view of a control panel window on the screen.

(Fig. 21)

A diagram, partly in plan view, to explain the correspondence of the classes to which the objects belong with the constituent elements of the digital VTR control panel object.

(Fig. 22)

A diagram to explain the production of a play button object.

(Fig. 23)

Flowcharts of the routines for activation of the icon for the digital VTR and for start of a playing operation.

(Fig. 24)

A plan view of a multimedia controller window on the screen after the user has selected the control mode in the icon display for the digital VTR.

(Fig. 25)

A diagram of the relationship between the structure of the deputy "data input to digital VTR" object and the data for describing the object.

(Fig. 26)

A diagram of the relationship between the structure of the deputy "data output from digital VTR" object and the data for describing the object.

(Fig. 27)

A diagram of the structure of the "data input to digital VTR" object.

(Fig. 28)

A diagram of the structure of the "data output from digital VTR" object.

(Fig. 29)

A diagram of the relationship between the structure of the digital VTR control panel object of panel class and the data for describing the object.

(Fig. 30)

Top views of the control panels on the display of the multimedia controller in operative positions.

(Fig. 31)

A block diagram of the structures of objects in the multimedia controller, VTR type device and digital camera type device with the states of connections of the

devices.

(Fig. 32)

A flowchart of the routine for the operation of the input/output data checking means.

(Fig. 33)

A flowchart of the routine for the input/output data checking means to operate when a link is set up from the digital camera to the digital VTR.

(Fig. 34)

A top view showing the design of a warning panel.

(Fig. 35)

Block diagrams of the components of the "methods" storage and the values of the internal data of the deputy "output from digital camera" object and the components of the "methods" storage and the values of the internal data of the deputy "input to digital VTR" object.

(Fig. 36)

A diagram of the format of motion picture data.

(Fig. 37)

A block diagram of the structure of the internal components of the digital VTR type device.

(Fig. 38)

A top view to explain how to use a user interface by the connection constructor window.

(Fig. 39)

A block diagram of the structure of objects

in the interior of the multimedia controller in the second embodiment of the invention.

(Fig. 40)

A block diagram showing the process for switching messages in the display means for the connection constructor window of the connection constructor object.

(Fig. 41)

A block diagram of the internal structure of the deputy "output from digital camera" object.

(Fig. 43)

A block diagram of the internal structure of the digital VTR-Camera controller object.

(Fig. 44)

Block diagrams of the updated values of the internal data of the objects after an outgoing link has been set in from the digital camera to the digital VTR.

(Fig. 45)

Tables to explain a list of file types available from the digital camera and a list of acceptable file types to the digital VTR.

24.04.95

(Name of Document)

Drawings

(Fig. 1)
Logic of Connection of Multimedia Controller with multimedia devices

1: Multimedia Controller

2: Multimedia Device (Digital Camera)
2: Multimedia Device (Printer)
2: Multimedia Device (Digital FAX)
2: Multimedia Device (Digital Copier)
2: Multimedia Device (Digital VTR)
2: Multimedia Device (CD Player)

(Fig. 2)
Physical configurations of connecting Multimedia Controller to Multimedia Devices

a) Daisy Chain Line
    1: Multimedia Controller
    2: Multimedia Device

b) Star Configuration
    1: Multimedia Controller
    2: Multimedia Device

c) Multipoint Line
    1: Multimedia Controller
    2: Multimedia Device

(Fig. 3)
Internal Structure of Object Embedded Multimedia Device

10: external Bus
15: Multimedia Data
16: Driver for Mechanical System
17: Mechanisms & Motors
18: Driver for Electrical System
19: Electrical Circuits, Indicators & Switches

(Fig. 4)

Internal Structure of Object Embedded Multimedia Controller

25: Multimedia Data Filing
26: Display Controller
27: Display
28: Driver for Electrical System
29: Electrical Circuits, Indicators & Switches
30: External Bus

(Fig. 5)
Hierarchy of Systems in Multimedia Controller

50: Hardware
53: Common Class Library
54: C Function
55: Specific Class Library
56: Application for Controlling Multimedia Devices

(Fig. 6)
Hierarchy of Systems in Multimedia Device

57: Hardware
59: Specific Class Library
60: C Function
61: Application for Multimedia Device

(Fig. 7)
1: Multimedia Controller
    205: System Director Object

2: Multimedia Device
    1061: File for Describing Deputy Multimedia Device Object
        1062: Section for Describing "Control Panel for Multimedia Device" Object
        1063: Section for Describing Deputy "Data I/O of Multimedia Device" Object
    1064: Multimedia Device Object
        1065: "Controller in Multimedia Device" Object
        1066: "Data Input to Multimedia Device" Object
        1067: "Data Output from Multimedia Device" Object

(Fig. 8)
1: Multimedia Controller
    205: System Director Object

    1068: Deputy Multimedia Device Object
        1069: "Control Panel for Multimedia Device" Object
        1070: Deputy "Data Input to Multimedia device" Object
        1071: Deputy "Data Output from Multimedia Device" Object

2: Multimedia Device
    1064: Multimedia Device Object
        1065: "Controller in Multimedia Device" Object
        1066: "Data Input to Multimedia Device" Object
        1067: "Data Output from Multimedia Device" Object

(Fig. 9)
1081: Class Library
- 1079: First Class
  - 1080: Class Definition Part
  - 1081: Class Method Table
  - 1082: Code Part
    - 1083: 1st Function Code
    - 1084: k-th Function Code
- 1085: p-th Class

(Fig. 10)
234: Object
- 235: Portion for Internal Data
  - 236: 1st Internal Data
  - 237: 2nd Internal Data
  - 238: n-th Internal Data
- 239: Portion for Methods
  - 240: 1st Data Processing Means
  - 241: 2nd Data Processing Means
  - 242: m-th Data Processing Means
- 243: Class Method Table
  - 240: 1st Data Processing Means
  - 241: 2nd Data Processing Means
  - 242: 3rd Data Processing Means
- 244: Portion for Accommodating Pointer for Class Method Table
- 245: Message Communicating Means; Message
- 246: Processing & Retrieving Means
- External Data

(Fig. 11)
205: System Director Object
- 342: Processing & Retrieving Means
- 1061: File for Describing Deputy Multimedia Device Object
- 1072: Portion for Accommodating Pointer for Class Method Table
- 1073: Class Method Table for System Director
  - 1047: Deputy Multimedia Device Object Forming means
  - 343: Input/output Data Checking Means
  - 380: Application Object Forming Means
- 1074: Message Communicating Means; Message
- 1075: Portion for Methods
  - 1047: Deputy Multimedia Device Object Forming means
  - 343: Input/Output Data Checking Means
  - 380: Application Object Forming Means
- 1076: Portion for Internal Data
  - 1077: Object ID
  - 1078: Object Registration Data
  - 344: Current Link Tracking Data

(Fig. 12)
247: Section for Describing Control Panel Object
  248: 1st Database for Describing Object
    250: Data for Recognizing Object
      251: Class Name
      252: Object ID
      253: Superior Object ID
    254: Data for Drawing Object
      255: 1st Data for Drawing Object
        256: Data for Location & Size
        257: Data for Pattern & Color
        258: Object Image
      259: j-th Data for Drawing Object
    260: Data for Object Link
      261: 1st Data for Object Link
        262: Relational Object ID
        263: Message for Transmission
      264: k-th Data for Object Link
  249: i-th Database for Describing Object

(Fig. 13)
650: File for Describing Deputy Data I/O Object
  651: 1st Data for Deputy Data Input Object
    652: Object ID
    653: Relational Data Input Object ID
    654: Compatible File Type List
  655: m-th Data for Deputy Data Input Object
  659: 1st Data for Deputy Data Output Object
    660: Object ID
    661: Relational Data Output Object ID
    662: Compatible File Type List
  663: n-th Data for Deputy Data Output Object

(Fig. 14)
1: Multimedia Controller
  205: System Director Object

203: Digital VTR
  206: Digital VTR Object
    207: Digital VTR Controller Object
    208: "Data Input to Digital VTR" Object
    209: "Data Output from Digital VTR" Object

  210: File for Describing Deputy Digital VTR Object
    211: Section for Describing "Control Panel for Digital VTR" Object
    212: Section for Describing Deputy "Data Output from Digital VTR" Object

(Fig. 15)
207: Digital VTR Controller Object
  1009: Portion for Accommodating Pointer for Class Method Table
  1010: Message Communicating Means
  1011: Processing & Retrieving Means
  1012: Portion for Methods

1019: Reproducing Means
1020: Recording Means
1015: Portion for Internal Data
204: Object ID
1016: Tape Running Speed
1017: Current Footage
1018: Class Method Table for Controller Class
1019: Reproducing Means
1020: Recording Means

(Fig. 16)
636: Connect Digital VTR to Network
637: System Director Object Detects When Digital VTR is Connected
638: System Director Object issues Device ID to Digital VTR
639: System Director Object Loads File for describing Deputy Digital VTR Object
640: System Director Object Generates Deputy Digital VTR Object Based on Information from the File
641: Deputy Digital VTR Object Displays Icon (Mini Panel) for Digital VTR in System Controller Window
642: Wait for Instruction from User

(Fig. 17)

(Fig. 18)
1: Multimedia Controller
205: System Director Object
220: Deputy Digital VTR Object
221: Deputy "Control Panel for Digital VTR" Object
222: Deputy "Data Input to Digital VTR" Object
223: Deputy "Data Output from Digital VTR" Object
203: Digital VTR
206: Digital VTR Object
207: Digital VTR Controller Object
208: "Data Input to Digital VTR" Object
209: "Data Output from Digital VTR" Object

(Fig. 19)
(Fig. 20)
(Fig. 21)
221: Panel Class: Control Panel for Digital VTR
285: Menu Class: Panel View Option Menu
286: Form Class: Timer Counter
288: Button Class: Rewind Button
289: Button Class: Reverse Play Button
290: Button Class: Pause Button
291: Button Class: Fast Feed Button
292: Button Class: Stop Button
293: Button Class: Recording Button
294: Button Group Class: Selection of Control Modes
295: Radio Button Class: Default Button
296: Radio Button Class: Advanced Button

(Fig. 22)

612: Play Button Object
- 613: Portion for Accommodating Pointer for Class Method Table
- 614: Message Communicating Means
- 615: Processing & Retrieving Means
- 616: Portion for Methods
  - 626: Button Initializing Means
  - 627: Button Drawing Means
  - 628: Click Response Means
- 620: Portion for Internal Data
  - 621: Object ID
  - 622: Data for Button State
  - 623: Drawing Parameter
  - 624: Link Data
- 625: Button Class Method Table for Button Class
  - 626: Button Initializing Means
  - 627: Button Drawing Means
  - 628: Click Response Means

625: Button When Not Pressed
626: Button When Pressed
297: Data for Object Recognition
- 298: Class Name: Button Class
- 299: Object ID: ID=7 Play Button
- 300: Super Object ID; ID=1 VTR Control Panel

601: 1st Data for Drawing Object
- 602: Data for Location & Size
- 603: Data for Pattern & Color
- 604: Object Image

605: 2nd Data for Drawing Object
- 606: Data for Location & Size
- 607: Data for Pattern & Color
- 608: Object Image

609: Object Link Data
- 610: Link Terminal Object ID ; ID of VTR Control Object
- 6121: Message to Transmit; Play

(Fig. 23)
643: User Double Clicks on Icon of Digital VTR
644: "Control Panel for Digital VTR" Object Presents Display of Control Panel for Digital VTR
645: Wait for User's Action
646: Click on Play button
647: "Control Panel for Digital VTR" Object Sends "Play" Message to Digital VTR Controller Object
648: Digital VTR Controller Object Activates Reproducing Means
649: Start Play Mode of Digital VTR

(Fig. 24)

(Fig. 25)

222: Deputy "Data Input to Digital VTR" Object
- 668: Portion for Accommodating Pointer for Class Method Table
- 669: Message Communicating Means
- 670: Processing & Retrieving Means
- 671: Portion for Methods
  - 680: Means for Initializing Deputy Data Input Object
  - 681: Link Data Updating Means
  - 678: Compatible File Type Replying Means
- 674: Portion for Internal Data
  - 675: Object ID
  - 676: Relational Data Input Object ID
  - 677: Compatible File Types
  - 1006: Link Data
- 679: Class Method Table For Deputy Data Input Class
  - 680: Means for Initializing Deputy Data Input Object
  - 681: Link Data Updating Means
  - 678: Compatible File Type Replying Means
- 682: Deputy Data Input Object Data
  - 683: Object ID
  - 684: Relational Data Input Object ID
  - 685: Compatible File Type List

(Fig. 26)

223: Deputy "Data Output from Digital VTR" Object
- 690: Portion for Accommodating Pointer for Class Method Table
- 691: Message Communicating Means
- 692: Processing & Retrieving Means
- 693: Portion for Methods
  - 694: Means for Initializing Deputy Data Input Object
  - 695: Means for Updating Link Data
  - 700: Means for replying Compatible File Type
- 696: Portion for Internal Data
  - 697: Object ID
  - 698: Relational Data Output Object ID
  - 699: Compatible File Types
  - 688: Link Data
- 1048: Class Method Table for Deputy Data Output Class
  - 694: Means for Initializing Deputy Data Output Object
  - 695: Means for Sending Data Input Command
  - 700: Means for replying Compatible File Type
- 1001: Deputy Output Object Data
  - 1002: Object ID
  - 1003: Relational Data Output Object ID
  - 1004: Compatible File Type List

(Fig. 27)
208: "Data Input to Digital VTR" Object
- 1022: Portion for Accommodating Pointer for Class Method Table
- 1023: Message Communicating Means
- 1024: Processing & Retrieving Means
- 1025: Portion for Methods
  - 1032: Means for Writing File
  - 1033: Means for Receiving Data
  - 686: Means for updating Link data
- 1028: Portion for Internal Data
  - 1029: Object ID
  - 1030: Link Data
- 1031: Class Method Table for Data Input Class
  - 1032: Means for Writing File
  - 1033: Means for Receiving Data
  - 686: Means for Updating Link Data

(Fig. 28)
209: "Data Output from Digital VTR" Object
- 1035: Portion for Accommodating Pointer for Class Method Table
- 1036: Message Communicating Means
- 1037: Processing & Retrieving Means
- 1038: Portion for Internal Data
  - 1045: Means for reading Data
  - 1046: Means for Transmitting Data
  - 687: Means for Updating Link Data
- 1041: Portion for Internal Data
  - 1042: Object ID
  - 1043: Link Data
- 1044: Class Method Table for Data Output Class
  - 1045: Means for Reading Data
  - 1046: Means for Transmitting Data
  - 687: Means for Updating Link Data

(Fig. 29)
221: "Control Panel for Digital VTR" Object
- 1401: Portion for Accommodating Pointer for Class Method Table
- 1402: Class Method Table for Panel Class
  - 1403: Means for initializing Panel
  - 1404: Means for Drawing Panel Drawing
  - 1405: Click Response Means
- 1408: Portion for Methods
  - 1403: Means for Initializing Panel
  - 1404: Means for Drawing Panel Drawing
  - 1405: Click Response Means
- 1410: Portion for Internal Data
  - 1411: Object ID
  - 1412: Data for Panel State
  - 1413: Drawing Parameters
- 1414: Data for Object Recognition
  - 1415: Class Name; Button Class
  - 1416: Object ID; ID = 1 Digital VTR Control Panel

1417: Super Object ID
1418: 1st Data for Drawing Object
1419: Data for Location & Size
1420: Data for Pattern & Color
1421: Object Image
1422: 2nd Data for Drawing Object
1423: Data for Location & Size
1424: Data for Pattern & Color
1425: Object Image
1426: Icon Image
1427: Control Panel for Digital VTR (Frame)

(Fig. 30)
Method of selecting a link between devices on the display of the multimedia controller.
(a)
(b) How to set up a link from digital camera to digital VTR when to do dubbing. The buttons which are rendered inaccessible when the link is set up are grayed out.

(Fig. 31)
Structure of Objects within Multimedia Controller According to 1st Embodiment.

322: Multimedia Controller
323: System Director Object
324: Input/Output Data Checking means
336: Communicating Means
335: Deputy Device Object Storage
326: Deputy Digital VTR Object
326: Deputy "Input to Digital VTR" Object
328: Control Panel Object
327: Button Object
331: Deputy Digital Camera Object
330: Deputy "Output from Digital Camera" Object
333: Control Panel Object
332: Button Object
337: LAN
338: Digital VTR Device
340 "Controller for Digital VTR Device" Object
406: Communicating Means
408: "Data Input to Digital VTR Device" Object
410: "Data Output from Digital VTR Device" Object
339: Digital Camera Device
341: "Controller for Digital Camera Device" Object
407: Communicating Means
409: "Data Output from Digital Camera Device" Object
411: "Data Input to Digital Camera Device" Object

(Fig. 32)
Flowchart of Routine for Operation of Input/Output Data Checking Means:

System Director Object Receives Message of Link from Device A to Device B. Processing and Retrieving means

in System Director Object selects Input/Output Data Checking means from Method Storage to activate.

S1: Inquire of Deputy "Output from Device A" Object about Available File Types
S2: Inquire of Deputy "Input to Device B" Object about Combinations of Acceptable File Types with Priority Order
S3: Check Compatibility of Files Types between Both Devices
Determine File Type According to Priority Order
S4: Is There Compatible File Type ?
S5: Has Most Compatible File Type Same Data Attribute as That of Most Prior Field Type ?
S6: Determine That Link in Question is Valid
Save Information of Fact That Valid Link Has Been Set in from Device A to Device B and of File Type for This Link in Current Link Data Storage
S7: Inform Message Source Object, Deputy "Output from Device A" object and Deputy "Input to Device B" Object of Fact That Valid Link Has Been Set in from Device A to Device B and of File Type for This Link.
S8: Determine That Link in Question is Invalid
Request Linked Message Source Object for Demanding Warning Display to Present That Data are Impossible to Transfer
S9: Warn Message Source Object
Wait for Continue or Cancel Message
S10: Continue ?
S11: Inform Message Source Object of Link Being Invalid.

(Fig. 33)
Flowchart of Routine for Operation of Input/Output Data Checking means when Link is set up from Digital Camera to Digital VTR.

System Director Object Receives Message of Having Linked Camera to VTR from Window Server;
Processing and Retrieving Means Activates Input/Output Data Checking Means

S1: Inquire of Deputy "Output from Digital Camera" Object about Available File Type
S2: Inquire of deputy "Input to Digital VTR" Object about Acceptable Field Types with Priority Order
S3: Check Compatibility of File Type Between Both Devices
Determine File Type According to Priority Order of Deputy "input to digital VTR" Object.
S4: Is There Compatible File Type ?
S5: Has Compatible File Type Same Data Attribute as That of Most Prior File Type ?
S6: Save Information of Fact That Valid Link Has Been Set in from Digital Camera to Digital VTR and of File type for this Link in Current Link Tracking Data
S7: Inform Deputy "Output from Digital Camera" Object and Deputy "Input to Digital VTR" Object of Fact That

Link has Been Set in from Digital Camera to Digital VTR and of File Type for this Link
S8: Request Window Server for Presenting Warning Display That Data are Impossible to Transfer
S9: Request Window Server for Presenting Warning Display with "YES" and "CANCEL" Buttons Demanded to Push
S10
S11:
S12: Has "YES" or "CANCEL" Been Pushed ?

(Fig. 34)
Warning Panel
Audio & Movie Data are Impossible to Dub
Do You Dub Movie Data ?

(Fig. 35)
(a) Method Storage Part and Internal Data Storage Part of Deputy "Output from Digital Camera" Object
Method Storage Part
Initializing Means for Deputy Output Object
357: Link Data Updating Means
358: Available File Type Response Means
Internal Data Storage Part
Object ID = 10
Relational Data Output Object ID = 110
359: Available File Type List = Table 1
Link Data = Data Destination Object ID = 120
Most Compatible Format for This Link = Movie 2

(b) Method Storage Part and Internal Data Storage Part of Deputy "input to Digital VTR" Object
Method Storage Part
Initializing Means for Deputy Input Object
360: Link Data Updating Means
361: Acceptable Field Type Response Means
Internal Data Storage Part
Object ID = 20
Relational Data Input Object ID = 120
Acceptable File Type List = Table 2
Link Data = Data Source Object ID = 110
Data Format for This Link = Movie 2

(Fig. 36)
Time-Serial Data of Movie 2 in Real Time Fashion
364: Header
365: Boundaries between Possible Minimum Units in which to Edit Data
Time Stamp

(Fig. 37)
Internal Structure of Digital VTR Device
Communicating Means
367: System Controller
368: Tape Deck
369: Magnetic Tape Recording Medium

(Fig. 38)
User Interface for Connection Constructor Window

(Fig. 39)
Structure of Object Within Multimedia Controller According to 2nd Embodiment

384: Multimedia Controller
385: System Director Object
386: Application Object Forming Means
LOAD; Object Generation
387: Class Storage Part
Window Class
Button Class
388: Application Class Storage Part
Connection Constructor Class
389: Application Object Work Area
390: Connection Constructor Object
391: Connection Constructor Window Display Means
392: Connection Editing Means
401: Multimedia Device Object Storage Part
394: CD-Player Object
396: Control Panel Object
Deputy Data Output Object
396: Button Object
398: Amplitude Object
400: Control Panel Object
397: Deputy Data Input Object
399: Slider Object
402: Communicating Means
404: CD-Player Device
405: Amplifier Device

(Fig. 40)

Flowchart of Routine for Operation of Display means for Connection Constructor Window in Connection Constructor Object

Activate Display Means for Connection Constructor Window

S1: Request System Director Object for showing list of Connected Device Object IDs
Messaging; System Director Object; Transmit Data of List of Connected Device Object IDs
S2: Loop of Waiting for Data to be transmitted
S3: Demand Graphics Data of Icons to All Device Objects
Messaging; Object of Device 1, ... , Object of Device N; Each Device Transmits Graphics Data of Icon
S4: Loop of Waiting for Data to be transmitted
S5: Inquire of Input/Output Data Checking Object about Current Link Data
Messaging; Input/Output Data Checking Object

Transmit Current Link Data
S6: Loop of Waiting for Data to be Transmitted
S7: Display Connection Constructor Window in which, Based on Link Data between Any Two of Devices, Icons are Arrayed so That Link lines, as illustrated, do not intersect one another as far as possible,

(Fig. 41)

410: Deputy "Output from Digital Camera" Object
Message Communicating Means; Message
411: Processing and Retrieving Means
412: Method Storage Part
Initializing means for Deputy Output Object
413: Link Data Updating Means
414: Available File Type Response Means
415: Internal Data Storage Means
416: Object ID
417: Data Destination Object ID
418: Compatible File Type
419: Link Data

(Fig. 42)
420: "Data Output from Digital Camera" Object
Message Communicating Means; Message
421: Processing and Retrieving Means
422: Method Storage Part
423: Data Reading Means
424: Data Transmitting Means
425: Link Data Updating Means
426: Internal Data Storage Part
427: Object ID
428: Link Data

(Fig. 43)
429: "Controller for Digital VTR" Object
430: Message Communicating Means; Message
431: Processing and Retrieving Means
432: Method Storage Part
433; Reproduction Mode Setting Means
434: Record Mode Setting Means
435: Internal Data Storage Part
204: Object ID
436: Tape Running Speed
437: Tape Footage

(Fig. 44)
Internal Data Storage Part of Each Object after Link Has Been Set in from Digital Camera to Digital VTR

(a) Internal Data Storage Part of Deputy "output from digital Camera" Object
Object ID = 10
Relational Data Output Object ID = 110
Available File Type List = As in Table 1

Link Data = (Data Destination Object ID = 120
Most Compatible Format for This Link = Movie 2)

(b) Internal Data Storage Part of Deputy "Input to Digital VTR" Object
Object ID = 20
Relational Data Input Object ID = 120
Acceptable File Type List = As in Table 2
Link Data = (Data Source Object ID = 110;
Data Format for This Link = Movie 2)

(c) Internal Data Storage Part of "Output from Digital camera" Object
Object ID = 110
430: Link Data = (Data Destination Object ID = 120;
Most Compatible Format for this
Link = Movie 2)

(d) Internal Data Storage Part of "Input to Digital VTR" Object
Object ID = 120
431: Link Data = (Data Source Object ID = 110; Most
compatible format for This Link
= Movie 2)

(e) Internal Data Storage Party of System Director Object
Object ID = 0
Object Registered Data = (Deputy "Digital VTR" Object
(ID = 201), Deputy "Digital
VTR Object (ID = 202),
...)
Current Link Tracking Data = ((Link ID = 10 to 20;
Type = Movie 2))

(Fig. 45)
(a) Available File Type List from Digital Camera
Data Attribute; File Type

(b) Acceptable File Type List to Digital VTR (with Priority Order)
Data Attribute; File Types Numbered in Priority Order

24:04:95

(Name of Document)

Written Abstract

(Abstract)

(Object) To provide an environment which does not requires that any multimedia device be equipped with an especial software and in which the multimedia devices can be used transparently from another controllers through the LAN.

(Constitution) A plurality of multimedia devices and a control device therefor are connected in a network. The multimedia devices and the control device are possible to transmit or receive messages and data based on the object-oriented concepts from one to another through the network. In such a system, the aforesaid control device is associated with display means and pointing means. By the aforesaid display means, all the multimedia devices are displayed in the form of respective symbolic icons. Using the pointing means, the user sets up links between these objects, thereby establishing a message switching network to transfer data from one of the devices to another.

(Selected Figure) Fig. 30